

ROTEIRO DE EXECUÇÃO ATUALIZADO

Foram testados novamente 8 casos (sendo alguns tautologias e outros não), tanto no nosso programa quanto no cliente web divulgado no classroom da atividade. Para rodar um dos casos de teste providos, basta chamar a função *verify* com um parâmetro *i*, onde *i* é um número natural para qual existe um teste “test_i.txt” na pasta “testes”. Alternativamente, pode-se usar a função *verifyFormula* passando uma string como fórmula (usando o encoding de implicação como ‘:’, negação como ‘~’ e conjunção, disjunção e variáveis atômicas como normal), ou a função *main*, que chama o teste padrão. Todos os resultados deixando fórmulas ser V ou F tiveram resultados iguais a quando fórmulas falsas eram tratadas como Não fórmulas.

Os resultados obtidos nos primeiros testes foram:

```
ghci> :l main
[1 of 2] Compiling Main                ( main.hs, interpreted ) [Source
file changed]
Ok, one module loaded.
ghci> verify 1
Formula: ((P ∨ (Q ∧ R)) -> ((P ∨ Q) ∧ (P ∨ R)))

Arvore:

0: F: ((P ∨ (Q ∧ R)) -> ((P ∨ Q) ∧ (P ∨ R)))
1:   V: (P ∨ (Q ∧ R))
2:     F: ((P ∨ Q) ∧ (P ∨ R))
3:       F: (P ∨ Q)
4:         F: P
5:           F: Q
6:             V: P [[Folha]]
6:             V: (Q ∧ R)
7:               V: Q
8:               V: R [[Folha]]
3:         F: (P ∨ R)
4:           F: P
5:             F: R
6:               V: P [[Folha]]
6:               V: (Q ∧ R)
7:                 V: Q
8:                 V: R [[Folha]]

Tautologia: True
ghci> verify 2
Formula: (P ∨ ~P)

Arvore:
```

```
0: F: (P ∨ ¬P)
1:   F: P
2:   F: ¬P
3:   V: P  [[Folha]]
```

Tautologia: True

ghci> verify 3

Formula: ((P -> Q) ^ P) -> Q)

Arvore:

```
0: F: ((P -> Q) ^ P) -> Q)
1:   V: ((P -> Q) ^ P)
2:   F: Q
3:   V: (P -> Q)
4:   V: P
5:   F: P  [[Folha]]
5:   V: Q  [[Folha]]
```

Tautologia: True

ghci> verify 4

Formula: (a -> (a -> (b -> a)))

Arvore:

```
0: F: (a -> (a -> (b -> a)))
1:   V: a
2:   F: (a -> (b -> a))
3:   V: a
4:   F: (b -> a)
5:   V: b
6:   F: a  [[Folha]]
```

Tautologia: True

ghci> verify 5

Formula: (b -> (a ^ (b ∨ a)))

Arvore:

```
0: F: (b -> (a ^ (b ∨ a)))
1:   V: b
2:   F: (a ^ (b ∨ a))
3:   F: a  [[Folha]]
3:   F: (b ∨ a)
4:   F: b
```

5: F: a [[Folha]]

Tautologia: False

ghci> verify 6

Formula: $\sim(\sim P \wedge (Q \rightarrow \sim(\sim P \vee Q)))$

Arvore:

0: F: $\sim(\sim P \wedge (Q \rightarrow \sim(\sim P \vee Q)))$

1: V: $(\sim P \wedge (Q \rightarrow \sim(\sim P \vee Q)))$

2: V: $\sim P$

3: V: $(Q \rightarrow \sim(\sim P \vee Q))$

4: F: Q

5: F: P [[Folha]]

4: V: $\sim(\sim P \vee Q)$

5: F: $(\sim P \vee Q)$

6: F: $\sim P$

7: F: Q

8: F: P [[Folha]]

Tautologia: False

ghci> verify 7

Formula: $(\sim\sim P \rightarrow \sim(P \rightarrow \sim Q))$

Arvore:

0: F: $(\sim\sim P \rightarrow \sim(P \rightarrow \sim Q))$

1: V: $\sim\sim P$

2: F: $\sim(P \rightarrow \sim Q)$

3: V: $(P \rightarrow \sim Q)$

4: F: P

5: F: $\sim P$

6: V: P [[Folha]]

4: V: $\sim Q$

5: F: Q

6: F: $\sim P$

7: V: P [[Folha]]

Tautologia: False

ghci> verify 8

Formula: $(\sim((p \wedge (q \vee r)) \rightarrow (s \vee t)) \rightarrow u)$

Arvore:

0: F: $(\sim((p \wedge (q \vee r)) \rightarrow (s \vee t)) \rightarrow u)$

1: V: $\sim((p \wedge (q \vee r)) \rightarrow (s \vee t))$

```
2:      F: u
3:      F: ((p ^ (q v r)) -> (s v t))
4:      V: (p ^ (q v r))
5:      F: (s v t)
6:      F: s
7:      F: t
8:      V: p
9:      V: (q v r)
10:      V: q  [[Folha]]
10:      V: r  [[Folha]]
```

Tautologia: False